



Algorithme assurant l'insertion dynamique d'un processeur autour d'un reseau a diffusion et garantissant la coherence d'un systeme de numerotation des paquets global et reparti

Annick Le Coz, Hervé Le Goff, Michel Ollivier

► To cite this version:

Annick Le Coz, Hervé Le Goff, Michel Ollivier. Algorithme assurant l'insertion dynamique d'un processeur autour d'un reseau a diffusion et garantissant la coherence d'un systeme de numerotation des paquets global et reparti. [Rapport de recherche] RR-0173, INRIA. 1982. inria-00076385

HAL Id: inria-00076385

<https://inria.hal.science/inria-00076385>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



CENTRE DE RENNES

IRISA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tél. 954 90 20

Rapports de Recherche

N° 173

**ALGORITHME ASSURANT
L'INSERTION DYNAMIQUE
D'UN PROCESSEUR AUTOUR
D'UN RÉSEAU À DIFFUSION
ET GARANTISSANT
LA COHÉRENCE D'UN SYSTÈME
DE NUMÉROTATION DES PAQUETS
GLOBAL ET RÉPARTI**

Annick LE COZ
Hervé LE GOFF
Michel OLLIVIER

Novembre 1982

ALGORITHME ASSURANT L'INSERTION DYNAMIQUE
D'UN PROCESSEUR AUTOUR D'UN RESEAU A DIFFUSION
ET GARANTISSANT LA COHERENCE D'UN SYSTEME DE
NUMEROTATION DES PAQUETS GLOBAL ET REPARTI

Annick LE COZ, Hervé LE GOFF, Michel OLLIVIER

RÉSUMÉ

Ce document présente un algorithme assurant l'insertion dynamique d'un processeur sur un réseau local.

Considérant que chaque processeur actif gère une copie d'une variable répartie qui représente le numéro du prochain paquet à circuler sur le réseau, tout processeur qui désire s'insérer doit acquérir une copie de la variable répartie afin d'en assurer la cohérence.

Nous prenons en compte les propriétés suivantes des réseaux locaux :

- diffusion possible des paquets
- unicité du paquet en transit
- séquençement naturel garanti par le médium de transmission.

Après avoir présenté le contexte de l'étude, nous explicitons la démarche qui a conduit à construire cet algorithme. Puis nous le décrivons précisément.

L'algorithme a été validé par simulation puis implanté sur un communicateur du réseau local DANUBE.

SUMMARY

This paper focuses on dynamic insertion of a processor on a local data network.

We consider that each active processor manages one copy of a distributed and duplicated information and that this information represents the number of the next packet on the network. In each active processor, the number is actualized at each send or receive operation. To preserve the coherence of this information, each new processor must initialize its own copy before turning in active state.

We consider the next proprieties of local networks : broadcasting, natural sequencing on the cable.

First, we introduce the context of the problem. Secondly, we propose and describe precisely a solution solving the problem.

The correctness of this solution has been verified by simulation. This solution has been implemented in a DANUBE local network communicator.

```

*****
* *****
* *
* *   ALGORITHME ASSURANT L'INSERTION DYNAMIQUE   * *
* *
* *   D'UN PROCESSEUR AUTOUR D'UN RESEAU A DIFFUSION * *
* *
* *   ET GARANTISSANT LA COHERENCE D'UN SYSTEME DE  * *
* *
* *   NUMEROTATION DES PAQUETS GLOBAL ET REPARTI   * *
* *
* *****
*****

```

Annick LE COZ*, Hervé LE GOFF**, Michel OLLIVIER***

- * IRISA - Université de Rennes
- ** IRISA - INRIA
- *** IRISA - INSA de Rennes

TABLE DES MATIERES

1 - INTRODUCTION

2 - L'ALGORITHME D'INSERTION DYNAMIQUE

2.1 - Principes generaux

2.2 - Squelette de l'algorithme

2.3 - Extensions de l'algorithme

3 - CONCLUSION

REFERENCES

1 - INTRODUCTION

=====

Un critère de qualité pour un réseau informatique est la modularité: il doit être possible d'augmenter le nombre de sites connectés, la quantité et la qualité des services proposés sans perturber le fonctionnement du système global.

Ce papier présente un algorithme assurant l'insertion dynamique d'un processeur dans un environnement de réseau informatique.

L'expression "insertion dynamique" inclut deux principes:

- Un site doit pouvoir s'insérer sur le réseau à tout instant, même lorsque d'autres sites sont déjà en cours d'activité.

- L'insertion d'un site ne doit pas affecter les éventuels échanges d'information établis entre les sites préalablement connectés.

Dans certains cas, l'insertion dynamique d'un site sur un réseau informatique ne nécessite pas de procédure explicite d'insertion.

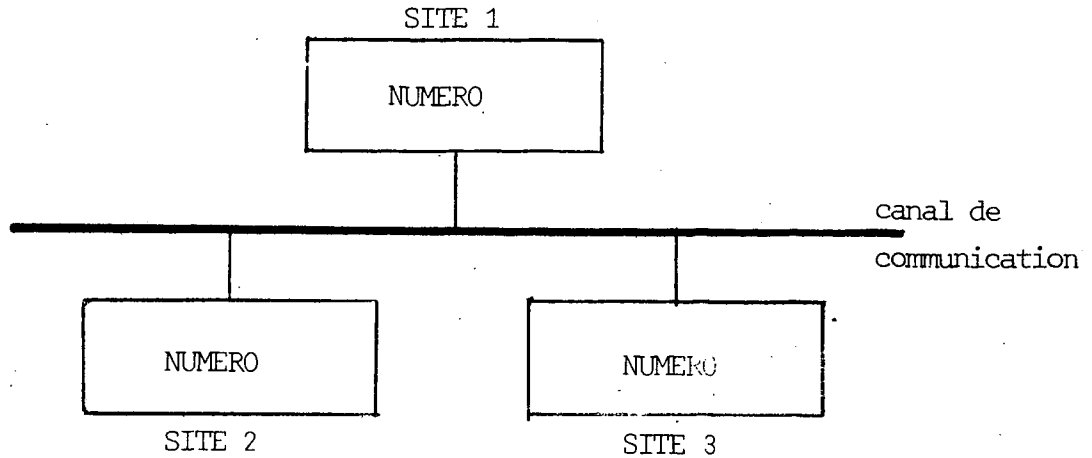
Dans d'autres cas, notamment lorsque chaque serveur de communication gère une copie d'une information globale au système, il est indispensable de mettre en oeuvre, lors de l'insertion d'un site, un traitement permettant à ce site d'acquérir l'état à jour de cette information afin de préserver la cohérence de ces copies multiples.

Le problème de la cohérence de copies multiples d'une information est étudié dans [Cornafion].

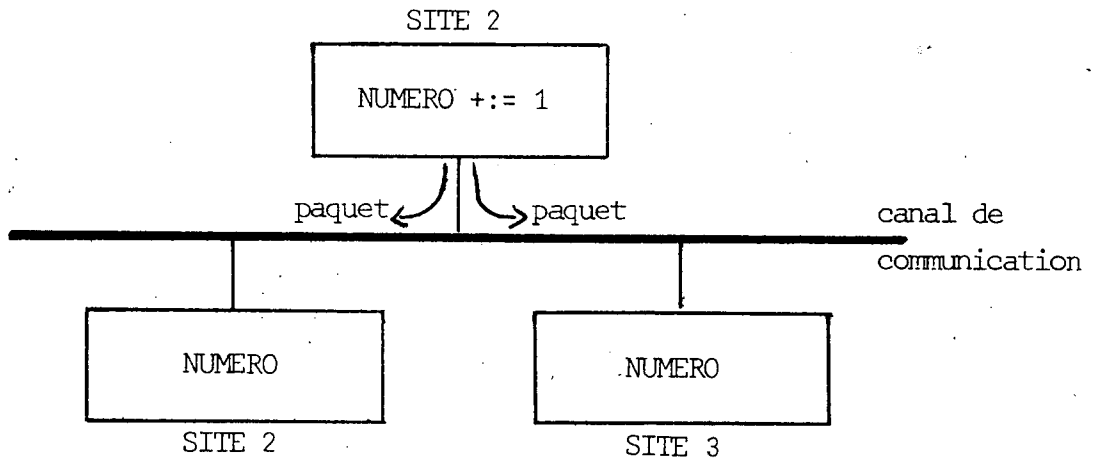
C'est dans ce contexte que s'intègre ce papier. En effet, l'algorithme décrit suppose l'existence d'un système de numérotation des paquets assurant l'ordonnancement global, unique et continu des paquets circulant sur le support de communication.

La fonction de numérotation des paquets est répartie sur chacun des serveurs de communication: chaque site gère une copie d'une variable globale appelée NUMERO représentant le numéro du prochain paquet à circuler.

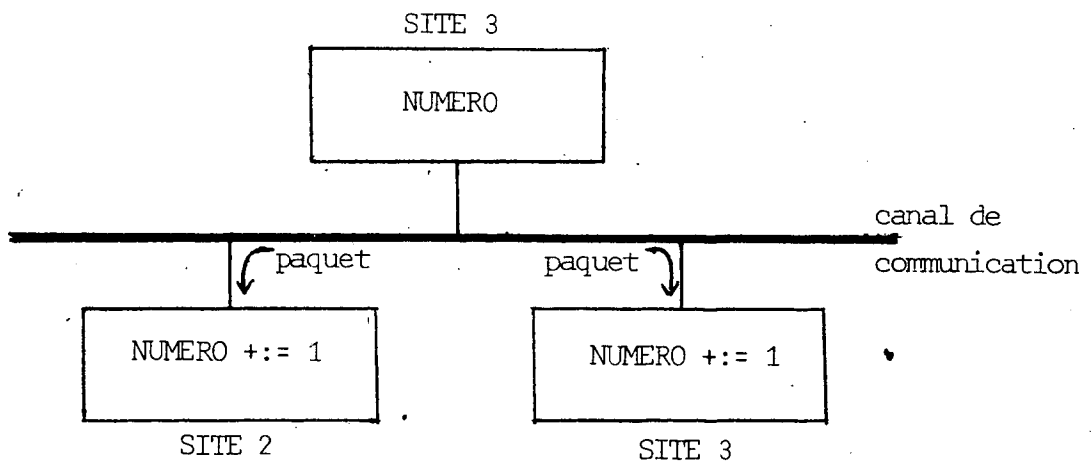
INTRODUCTION



a) Chaque site gère une copie de la variable NUMERO.



b) Le site 1 émet un paquet.



c) Réception du paquet sur les SITE 2 et SITE 3.

SCHEMA 1 : Evolution de la numérotation.

INTRODUCTION

L'évolution de la variable NUMERO sur un site s'effectue de la manière suivante [Schéma 1] :

- Si le site émet un paquet alors incrémenter NUMERO.
- Si le site reçoit un paquet (même erroné) alors incrémenter NUMERO.

-Dans tous les autres cas, NUMERO ne varie pas.
Ce sont donc uniquement les événements "transfert de paquet sur le canal en émission ou en réception" qui font évoluer la variable NUMERO.

Trois facteurs contribuent à maintenir la cohérence entre les états des différentes instances de la variable NUMERO:

- L'adressage systématiquement utilisé est la diffusion c'est-à-dire qu'un paquet émis par un site est transmis à chacun des autres sites connectés.

- A un instant donné, il existe au plus un paquet en transit sur le canal.

- On suppose qu'il n'y a pas de perte de paquet.

Ce système de numérotation est présenté dans [Gaucher-Cazalis].

Les hypothèses d'utilisation de l'algorithme décrit dans ce papier sont les suivantes:

H1 - L'information est véhiculée sur le support de transmission par paquets et on suppose qu'un paquet ne contient pas son propre numéro.

H2 - A partir de l'instant où un site au moins est connecté sur le réseau, il doit exister un trafic minimum sur le canal de communication. Par exemple, périodiquement, on peut imaginer que chaque site actif diffuse un message indiquant son état.

H3 - Chaque paquet transmis contient l'identification de son site émetteur.

H4 - Le temps de transfert d'un paquet sur le support de transmission est majoré. Cette hypothèse intervient lors de l'évaluation de la durée des différents time-out utilisés dans l'algorithme d'insertion dynamique.

H5 - Le canal de communication est fiable. Par conséquent, le taux d'erreurs de transmission est faible.

Avant de concevoir l'algorithme d'insertion dynamique, plusieurs objectifs ont été fixés:

INTRODUCTION

-Une tentative d'insertion sur le réseau ne doit pas être autorisée lorsque la charge du système est importante. En effet, dans ce cas, les conditions d'exploitation sont moins favorables: la restitution d'informations entre les différents sites est réalisée beaucoup plus lentement, les risques de collision de paquets augmentent...

Si dans une telle situation, on autorise l'insertion de nouveaux sites, le trafic s'intensifie ce qui a pour effet de réduire plus encore les performances du système.

L'initialisation d'un serveur de communication est donc considérée comme une action peu prioritaire par rapport aux échanges en cours.

-La simultanéité de tentatives d'insertion doit être résolue au mieux c'est-à-dire que l'algorithme doit être conçu de manière à ce que la simultanéité de plusieurs tentatives d'initialisation n'entraîne pas l'échec de chacune d'entre elles.

-Enfin, s'il est nécessaire de bloquer le trafic pendant la phase d'insertion d'un site, il faut minimiser le délai d'interruption des échanges.

On va maintenant passer à la description progressive de l'algorithme défini, en exposant au préalable en quels termes se pose le problème de l'insertion dynamique d'un serveur de communication.

2 - L'ALGORITHME D'INSERTION DYNAMIQUE

2.1 - Principes généraux

Le principe de l'insertion dynamique d'un site consiste, pour ce site, à mettre à jour son instance de la variable NUMERO de manière cohérente avec les instances des sites préalablement connectés, ceci afin de garantir la compatibilité de fonctionnement de ce serveur de transport avec celui des serveurs déjà actifs.

La difficulté résulte de l'hypothèse qu'un paquet ne porte pas en lui son propre numéro: lors de l'insertion sur le réseau, une simple écoute du trafic n'est donc pas suffisante pour acquérir la valeur à jour de la variable NUMERO. Le site concerné doit, par conséquent, formuler explicitement une demande d'acquisition de la valeur de cette variable (demande appelée D_INIT) qui recevra une réponse contenant la valeur NUMERO (réponse appelée R_INIT).

Concernant les primitives D_INIT et R_INIT, deux contraintes sont à respecter:

C1 - Les primitives D_INIT et R_INIT doivent être consécutives c'est-à-dire qu'il faut interdire la circulation d'un paquet sur le support de transmission entre l'émission de ces deux primitives.

C2 - La réponse R_INIT ne doit être générée que par un seul site.

Par ailleurs, quelques considérations quant à l'algorithme:
-L'algorithme d'insertion dynamique doit inclure le cas de l'initialisation du système (c'est-à-dire l'insertion du premier site sur le réseau).

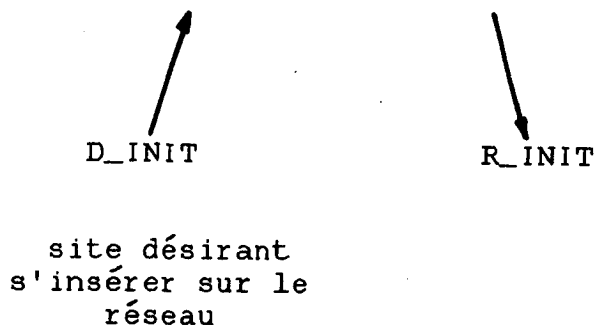
-L'algorithme doit être fiable: il doit résister aux erreurs de transmission ainsi qu'aux éventuelles défaillances de processeurs.

2.2 - Squelette de l'algorithme

Dans un premier temps, on va s'efforcer de montrer comment, à partir des principes de base de l'initialisation, on a élaboré un "squelette" d'algorithme.

* Le "cœur" de l'algorithme

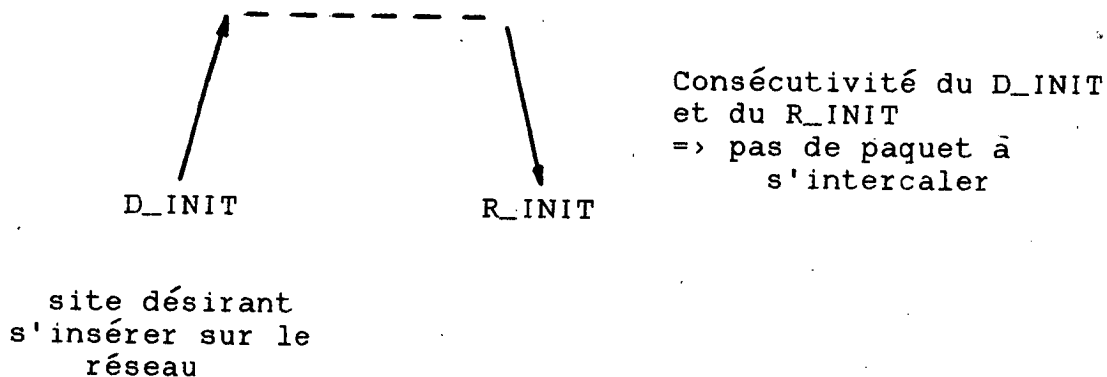
L'algorithme d'insertion dynamique d'un site s'articule autour de la séquence "demande d'acquisition, de la valeur NUMERO / réponse à cette demande", soit la séquence "D_INIT / R_INIT".



Schema 2 : LE COEUR DE L'ALGORITHME

* Le silence

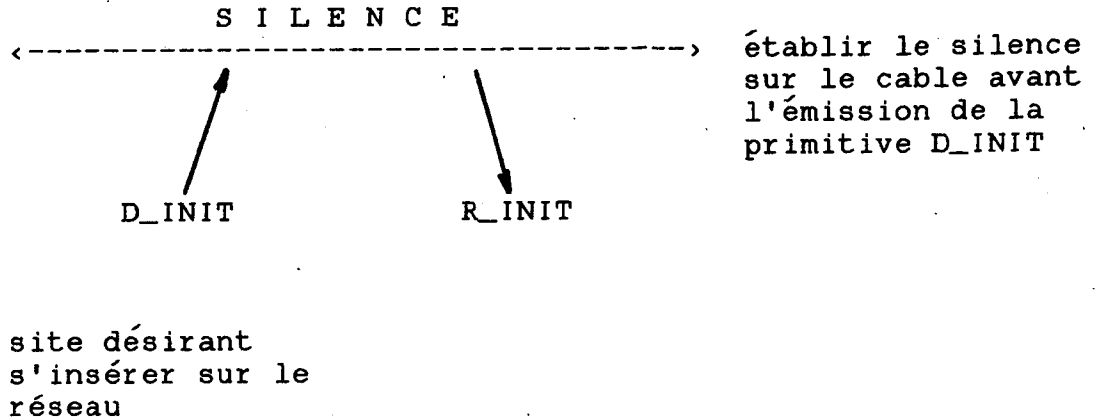
-La contrainte C1 impose que les primitives D_INIT et R_INIT soient consécutives (aucun paquet ne doit s'intercaler entre l'émission de ces deux paquets).



Schema 3

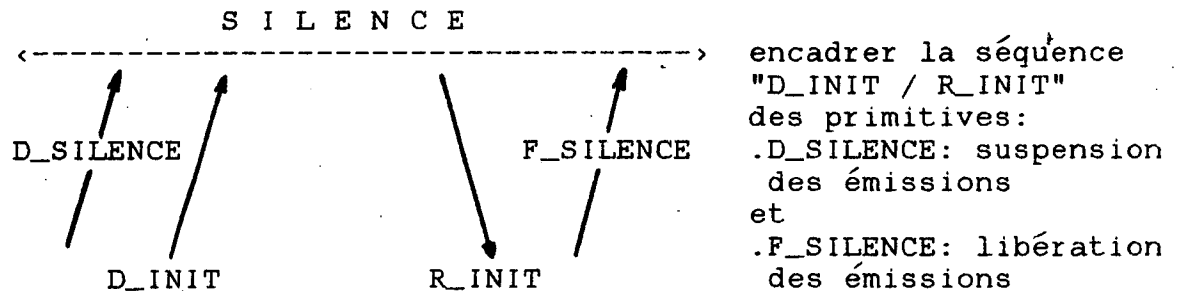
-L'idée consiste à établir le silence complet sur le support de transmission avant de diffuser le message D_INIT, puis à maintenir ce silence jusqu'à ce que la réponse R_INIT ait été reçue.

L'ALGORITHME D'INSERTION DYNAMIQUE



Schema 4

-La solution adoptée consiste à définir une primitive D_SILENCE qui sera diffusée avant l'émission de la demande D_INIT et qui sera interprétée par les sites déjà initialisés comme un ordre de suspension des émissions. Etant donné que, seul le serveur en cours d'initialisation est capable de juger à quel moment sa tentative d'insertion s'achève, il est fondamental que ce soit lui qui émette le signal de libération des émissions préalablement bloquées grâce à la primitive D_SILENCE. On a donc défini à cet effet la primitive F_SILENCE qui est interprétée par les sites initialisés comme un ordre de déblocage des émissions.



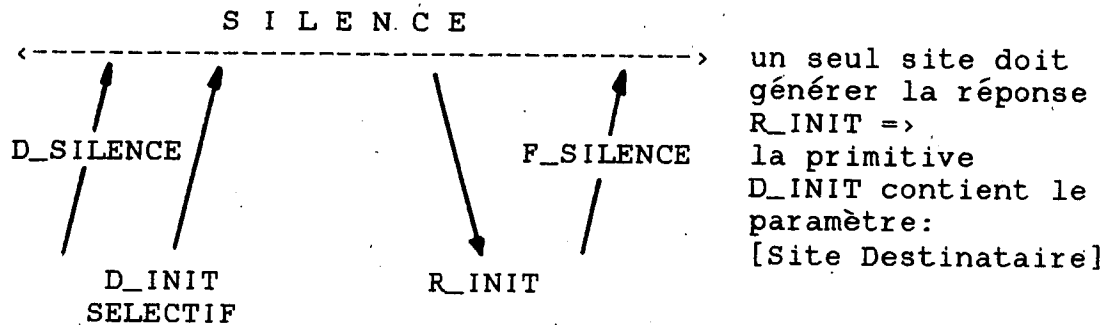
site désirant s'insérer sur le réseau

Schema 5

* L'écoute du réseau

-La contrainte C2 impose que la réponse R_INIT a la demande d'acquisition de la valeur NUMERO, D_INIT, ne soit générée que par un seul serveur de communication (celui-ci devant, bien entendu, être lui-même initialisé).

-L'idée consiste à rendre la primitive D_INIT sélective en y mentionnant l'identification d'un site destinataire qui sera chargé de renvoyer la réponse R_INIT.



site désirant s'insérer sur le réseau

Schema 6

-Ayant décidé de rendre le D_INIT sélectif, il reste encore à déterminer de quelle manière sera choisi le destinataire. Plusieurs solutions sont envisageables: on peut, par exemple, désigner le destinataire de la primitive D_INIT au hasard. On a opté pour une solution plus satisfaisante qui consiste à adresser la demande D_INIT à un site dont on a détecté l'activité.

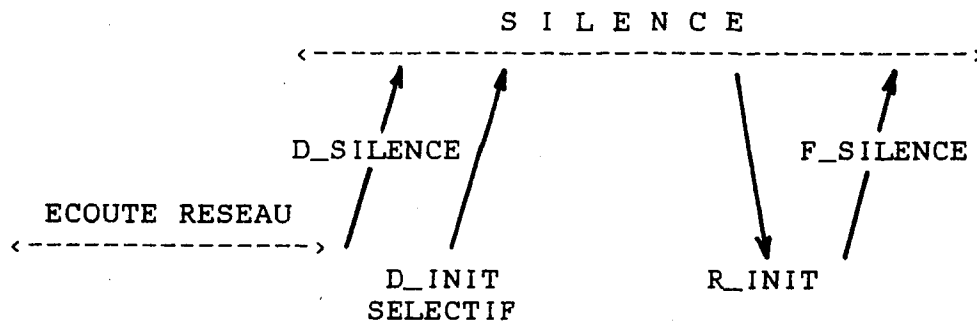
C'est pourquoi, on a introduit une phase d'écoute du réseau avant l'émission de la primitive D_SILENCE.

Pendant cette phase, le serveur en cours d'initialisation mémorise l'identification des sites dont il intercepte les paquets émis, en tant que destinataires potentiels de la primitive D_INIT.

Cette phase d'écoute du réseau permet de plus de résoudre le cas de l'initialisation du système: si au terme d'un délai T0 suffisamment long aucun paquet n'est reçu, le serveur en déduit qu'il est le premier à s'insérer sur le réseau; il positionne alors son instance de la variable NUMERO à zéro.

On voit ici l'intérêt de l'hypothèse énoncée dans l'introduction et qui exige l'existence d'un trafic minimum sur

le support de transmission dès l'instant où au moins un site est initialisé.



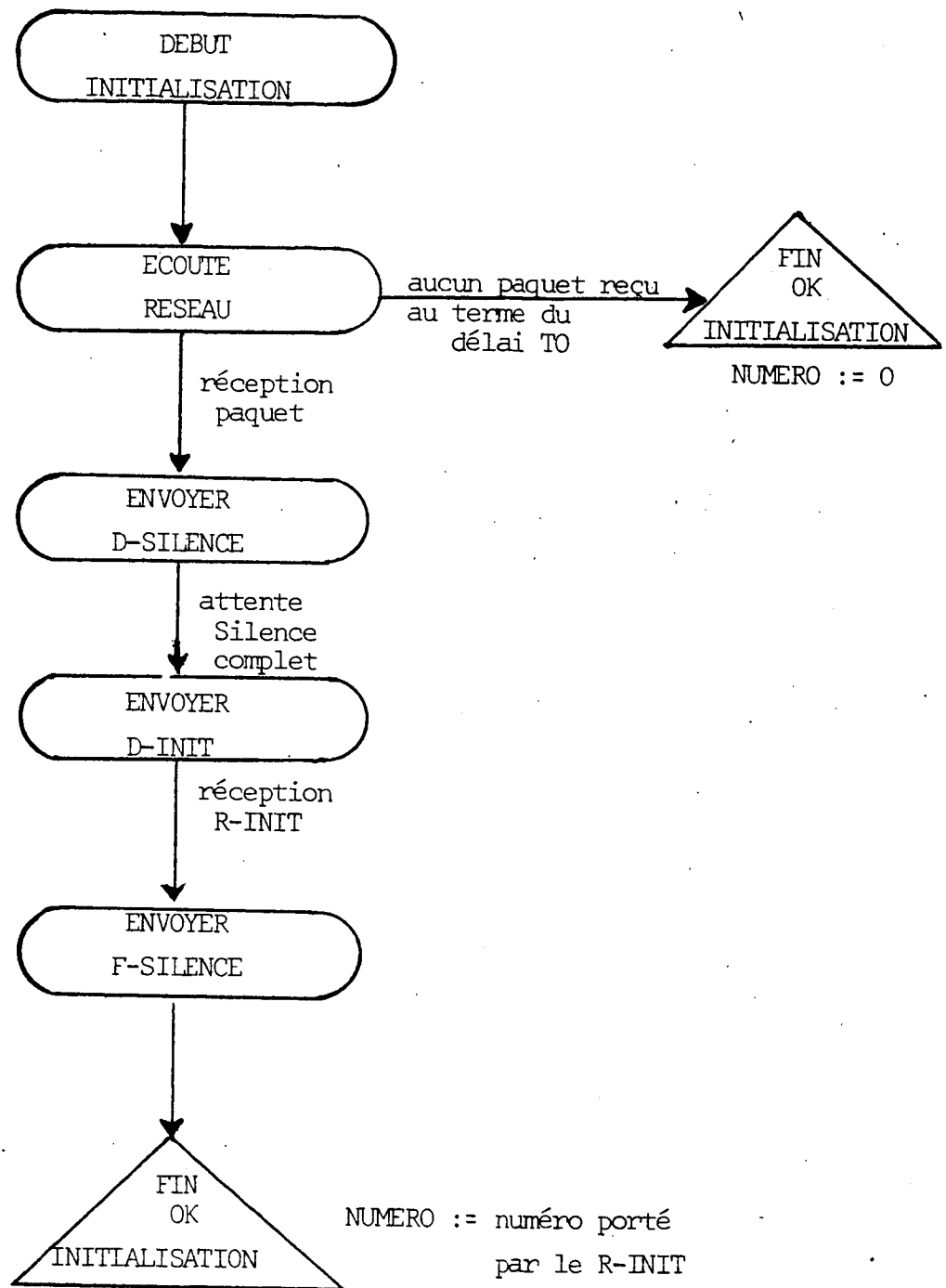
site désirant s'insérer sur le réseau

Schema 7: CHOIX DU DESTINATAIRE?

----- Ecoute du réseau avant l'émission
de D_SILENCE.
=> mémorisation des serveurs initialisés

L'ensemble de ces réflexions permettent alors de construire le squelette d'algorithme d'insertion dynamique présenté dans le [Schéma 8].

L'ALGORITHME D'INSERTION DYNAMIQUE



SCHEMA 8 : Le squelette de l'algorithme d'insertion dynamique.

Ce squelette d'algorithme correspond à une tentative d'insertion dynamique dans un environnement idéal et ne tient pas compte des propriétés de l'initialisation établies dans l'introduction.

Il faut donc greffer à cet algorithme des traitements destinés à :

-récupérer les erreurs de transmission ainsi que les
éventuelles défaillances de processeurs

Chacune des primitives D_SILENCE, D_INIT, R_INIT et F_SILENCE joue un rôle fondamental dans la démarche de l'initialisation. En particulier, les primitives D_SILENCE et F_SILENCE permettent d'établir puis de rompre le silence nécessaire à la formulation d'une demande d'insertion. Il est donc indispensable de mettre en oeuvre un traitement assurant la bonne réception de chacune de ces primitives sur tous les serveurs de communication connectés au réseau.

Par ailleurs, l'occurrence d'une erreur de transmission sur l'une des primitives D_INIT ou R_INIT peut mener le serveur en attente de la réponse R_INIT dans un état anormal.

Dans cette éventualité, il convient de prévoir au niveau de ce serveur un délai de garde au terme duquel il revient dans un état cohérent. On verra que deux solutions sont alors possibles :

-Le serveur formule une nouvelle demande d'initialisation, D_INIT.

-Il y a échec de la tentative d'insertion sur le réseau.

En ce qui concerne les défaillances de processeurs, il est important de remarquer qu'elles peuvent aussi bien affecter un serveur actif qu'un serveur en cours d'initialisation.

Un serveur en cours d'insertion doit donc considérer possible la défaillance du serveur auquel il adresse une primitive D_INIT. Dans ce cas, bien entendu, il ne reçoit pas la réponse R_INIT. Il faut donc, là-aussi, prévoir un délai de garde au terme duquel le serveur revient dans un état cohérent.

Par ailleurs, on a vu que lorsqu'un site actif reçoit une primitive D_SILENCE, il suspend ses émissions et ne les libère que sur réception de la primitive F_SILENCE. Si le serveur qui a soumis le paquet D_SILENCE est, peu après, victime d'une défaillance, la primitive F_SILENCE ne sera pas diffusée. Il est donc nécessaire dans l'algorithme d'envisager cette éventualité afin d'éviter le blocage infini du trafic.

-éviter l'insertion d'un site sur le réseau lorsque la charge du canal de communication est importante

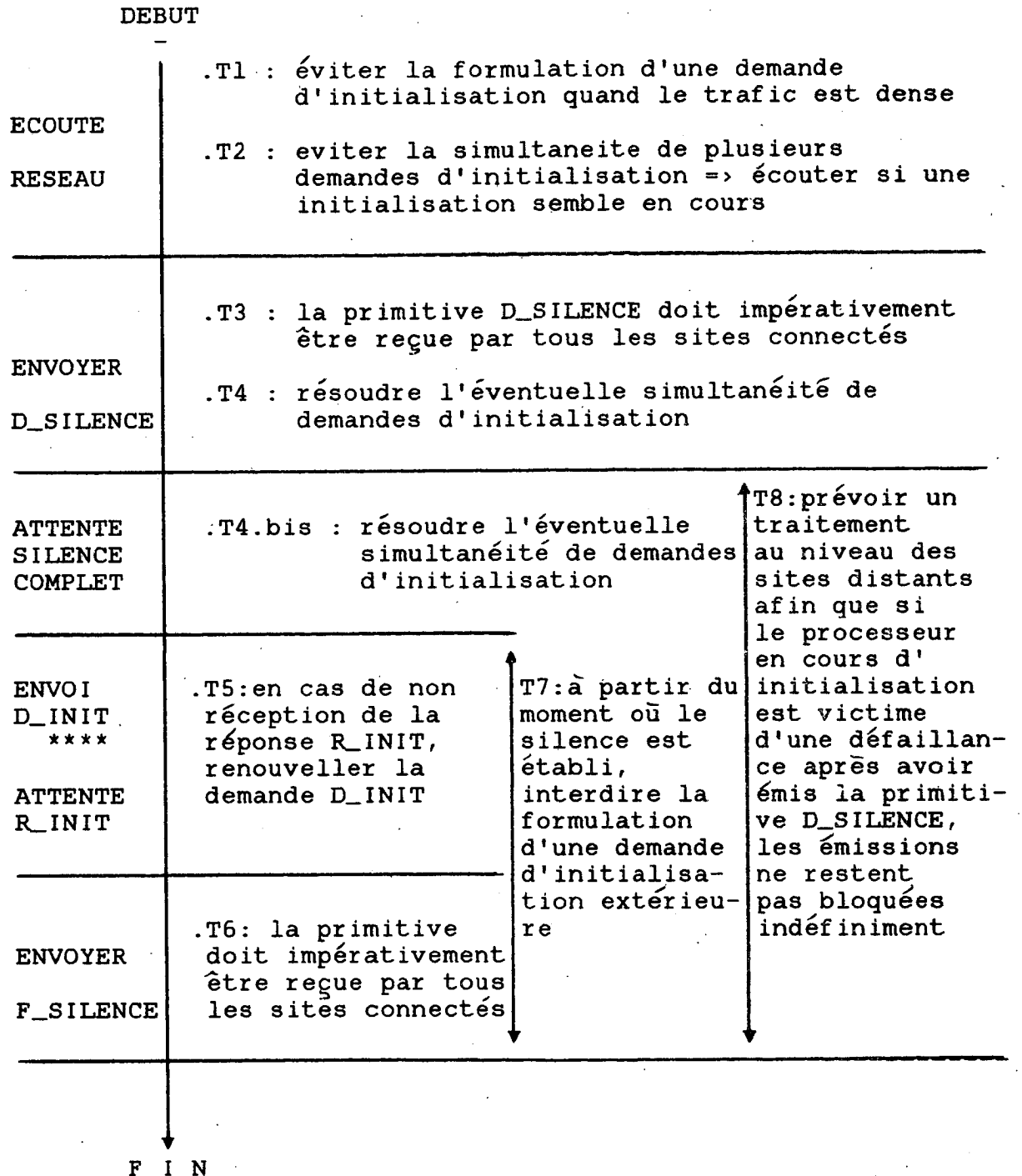
Il va de soi que ce problème doit être résolu avant la diffusion de toute information signalant une tentative d'insertion c'est-à-dire avant l'émission de la primitive D_SILENCE.

-minimiser les risques de simultanéité de soumission de demandes d'initialisation sur le canal de communication

Face à ce problème, il faut résoudre de manière satisfaisante cette simultanéité, c'est-à-dire faire en sorte que chacune des tentatives d'insertion impliquées soit menée à terme. Il est possible de minimiser les risques de simultanéité de soumission de demandes d'initialisation en écoutant pendant les phases précédant l'émission du D_SILENCE si une initialisation semble en cours (on analyse les paquets reçus). Si tel est le cas, le serveur renonce à formuler explicitement une demande d'initialisation et se met à l'écoute de la tentative d'insertion dont il a détecté l'existence.

En se basant sur le squelette d'algorithme élaboré, on a récapitulé chronologiquement, dans le [Schéma 9], l'ensemble des traitements à prévoir.

L'ALGORITHME D'INSERTION DYNAMIQUE



Schema 9 : REPARTITION DES TRAITEMENTS A PREVOIR

2.3 - Extensions de l'algorithme

-Traitements T1 et T2

Afin de traiter conjointement T1 et T2, on va scinder la phase "Ecoute Réseau" en deux phases:

a- La phase "Ecoute Réseau"

Le principal intérêt de cette phase est de déterminer l'état du système (vide ou non).

Au début de cette phase, on arme un time-out TO dont la durée est fonction de la période de génération des paquets destinés à assurer un trafic minimum dès l'instant où au moins un site est connecté.

-Si l'un des paquets D_SILENCE, D_INIT ou R_INIT est reçu, on en déduit qu'il existe une autre tentative d'insertion en cours. Le time-out TO est désarmé et le serveur se met à l'écoute de cette initialisation extérieure.

-On verra par la suite qu'il existe un cas où un serveur peut diffuser la primitive F_SILENCE alors que sa tentative d'insertion a échoué.

C'est pourquoi, si pendant la phase "Ecoute Réseau" un paquet erroné ou la primitive F_SILENCE est reçu, le serveur ne change pas d'état mais positionne un booléen signalant qu'il existe au moins un site actif (le time-out TO n'est pas désarmé).

-Si un paquet de données est reçu, le time-out TO est désarmé; le serveur mémorise l'identification du site émetteur de ce paquet et passe dans la phase "Ecoute Etat Sites Distants" présentée ci-dessous.

-Si lors du déclenchement du time-out TO, le booléen témoin de l'activité d'un site est positionné, le serveur reprend la phase "Ecoute Réseau" à son début et réarme le réveil TO. Sinon, il en déduit qu'il est le premier site à s'insérer sur le réseau: son instance de la variable NUMERO est initialisée à zéro.

b- La phase "Ecoute Etat Sites Distants"

Cette phase permet:

-D'une part, d'établir, par l'analyse des paquets reçus, s'il existe une initialisation extérieure en cours. Si tel est le cas (réception de l'une des primitives D_INIT, D_SILENCE ou R_INIT), le serveur renonce à formuler explicitement une demande d'initialisation afin d'éviter l'interférence de

plusieurs demandes et tente de profiter de cette initialisation extérieure pour réaliser sa propre insertion sur le réseau.

-D'autre part, d'éviter la formulation d'une demande d'initialisation quand le trafic est dense.

L'idée consiste à ne soumettre la primitive D_SILENCE qu'après avoir observé un temps de silence non négligeable sur le canal de communication.

A cette fin, on utilise un second time-out T01 géré de la manière suivante: le time-out est armé au début de la phase "Ecoute Etat Sites Distants". Si un paquet relatif à l'initialisation (autre que F_SILENCE) est reçu, le time-out T01 est désarmé et le serveur se met à l'écoute de l'initialisation extérieure en cours. A la réception de tout autre paquet, le réveil T01 est réenclenché (de plus, si le paquet reçu est un paquet de données, on note l'identification de son site émetteur).

De cette manière, l'objectif est atteint: la primitive D_SILENCE ne sera soumise que si l'on observe sur le support de transmission un silence de durée T01.

-Traitements T3 et T6

Les primitives D_SILENCE et F_SILENCE jouent un rôle essentiel dans la démarche d'insertion dynamique car ce sont elles qui ordonnent la suspension puis la libération des émissions des serveurs déjà initialisés. C'est donc grâce à elles qu'il est possible d'instaurer puis de rompre le silence sur le support de transmission, silence nécessaire à la formulation de la demande d'acquisition de la variable NUMERO.

C'est pourquoi, pour remédier aux éventuelles erreurs de transmission pouvant affecter l'une de ces primitives, on émet systématiquement une série de N primitives, N étant choisi de manière à faire tendre vers zéro la probabilité pour qu'un site ne reçoive correctement aucune primitive.

-Traitement T4

A ce stade, il ne s'agit plus de prévenir la simultanéité de plusieurs demandes d'initialisation mais de la traiter.

Cette situation correspond à l'évolution parallèle du processus d'initialisation dans au moins deux sites distincts.

Elle se caractérise de la manière suivante: pendant l'émission de sa série de primitives D_SILENCE, un serveur reçoit une primitive D_SILENCE extérieure.

Ce problème se résout en instaurant une "priorité" entre les demandes d'initialisation. Cette priorité se calcule par comparaison entre les identifications des sites impliqués, ce

qui sous-entend qu'une primitive D_SILENCE doit comporter le paramètre [identification du site émetteur]. Par exemple, on dira que la tentative d'insertion du site ayant l'identification de plus grande valeur est prioritaire. Ce serveur et lui seul poursuivra normalement sa tentative d'initialisation [INITIALISATION LOCALE ACTIVE]. Les autres serveurs impliqués se mettront à l'écoute de cette tentative d'initialisation prioritaire et essaieront d'intercepter le R_INIT [INITIALISATION LOCALE PASSIVE].

-Traitement T4.bis

Tout d'abord, il convient de justifier l'existence de la phase "Attente Silence Complet". En effet, elle peut sembler superflue car l'émission préalable de la série de primitives D_SILENCE a provoqué la suspension des émissions de tous les sites connectés.

Ce sont des considérations matérielles qui ont amené à introduire cette phase. En effet, on peut disposer d'un matériel ayant la caractéristique suivante: un paquet chargé dans le buffer émission sur le canal de communication n'est plus accessible.

Dans ce cas, la diffusion des primitives D_SILENCE ne suffit pas à garantir l'établissement du silence sur le support de transmission. Il faut prévoir en plus un délai de vidage des buffers émission déjà chargés lors de la réception du D_SILENCE.

Le traitement mis en oeuvre pour garantir l'établissement du silence complet est le suivant: après avoir émis ses N paquets D_SILENCE et au début de la phase "Attente Silence Complet", le serveur arme un time-out T02.

S'il reçoit un paquet avant le déclenchement de T02, il réenclenche ce réveil.

Pour garantir l'établissement du silence complet, il suffit donc d'évaluer le time-out T02 de manière à ce que l'existence d'un temps mort de durée T02 sur le canal implique que tous les buffers émission ont été vides.

Dans ces conditions, il est important de signaler qu'il est possible qu'un serveur se trouvant dans la phase "Attente Silence Complet" reçoive une primitive D_SILENCE dont l'émission a été retardée (par exemple, suite à une ou plusieurs collisions). Dans ce cas, le traitement exposé ci-dessus (traitement T4) est appliqué: si la primitive reçue correspond à une tentative d'insertion prioritaire, le serveur se met à l'écoute de cette initialisation extérieure; sinon, il poursuit normalement.

-Traitement T5

Pendant les phases précédant l'émission de la primitive D_INIT, on constitue une liste des serveurs initialisés dont on a détecté la présence (c'est-à-dire dont on a reçu un paquet de données).

Après avoir constaté l'établissement du silence complet sur le support de transmission (temps mort de durée T02), le serveur soumet la demande D_INIT (sélective). Il l'adresse au dernier serveur de transport mémorisé dans la liste des serveurs initialisés puis efface l'identification de ce serveur de la liste.

Le serveur arme alors un time-out T03. Dans le cas favorable, la réponse R_INIT est reçue avant le déclenchement de T03: le serveur initialise sa copie de la variable NUMERO avec la valeur mentionnée dans le paquet R_INIT; son insertion dynamique sur le réseau est réalisée.

Cependant, il se peut que le réveil T03 se déclenche sans que le serveur ait reçu la réponse R_INIT. Ce phénomène a trois origines possibles:

- réception erronée du D_INIT sur le site destinataire.
- réception erronée du R_INIT sur le site en attente de cette réponse.
- défaillance du site destinataire du D_INIT.

Dans une telle éventualité, le serveur doit, si possible, renouveler sa demande d'acquisition de la valeur NUMERO, D_INIT. Dans ce but, il consulte la liste des serveurs initialisés qu'il a constituée.

Si elle n'est pas vide, on réitère le même traitement.

Par contre, si cette liste est vide (c'est donc qu'il n'existe plus de destinataire possible pour le D_INIT), c'est l'échec de la tentative d'insertion mais, afin de ne pas prolonger la paralysie des échanges, le serveur diffuse le signal de libération des émissions, F_SILENCE.

Ceci explique la remarque faite précédemment signalant qu'un serveur peut émettre la primitive F_SILENCE alors que sa tentative d'insertion a échoué.

-Traitement T7

A partir du moment où le silence complet est établi sur le canal, on doit interdire la formulation d'une demande d'initialisation extérieure. Ceci est mis en oeuvre par l'établissement d'une relation d'ordre entre les différents time-out utilisés.

Certains des time-out employés sont facilement estimables:

- La valeur du time-out T03 est égale à deux fois le délai maximum de transfert d'un paquet de buffer émission à buffer réception (correspondant à la transmission du D_INIT puis du R_INIT) plus un délai supplémentaire nécessaire au traitement

de la primitive D_INIT sur le site destinataire et à la formulation de la réponse R_INIT associée.

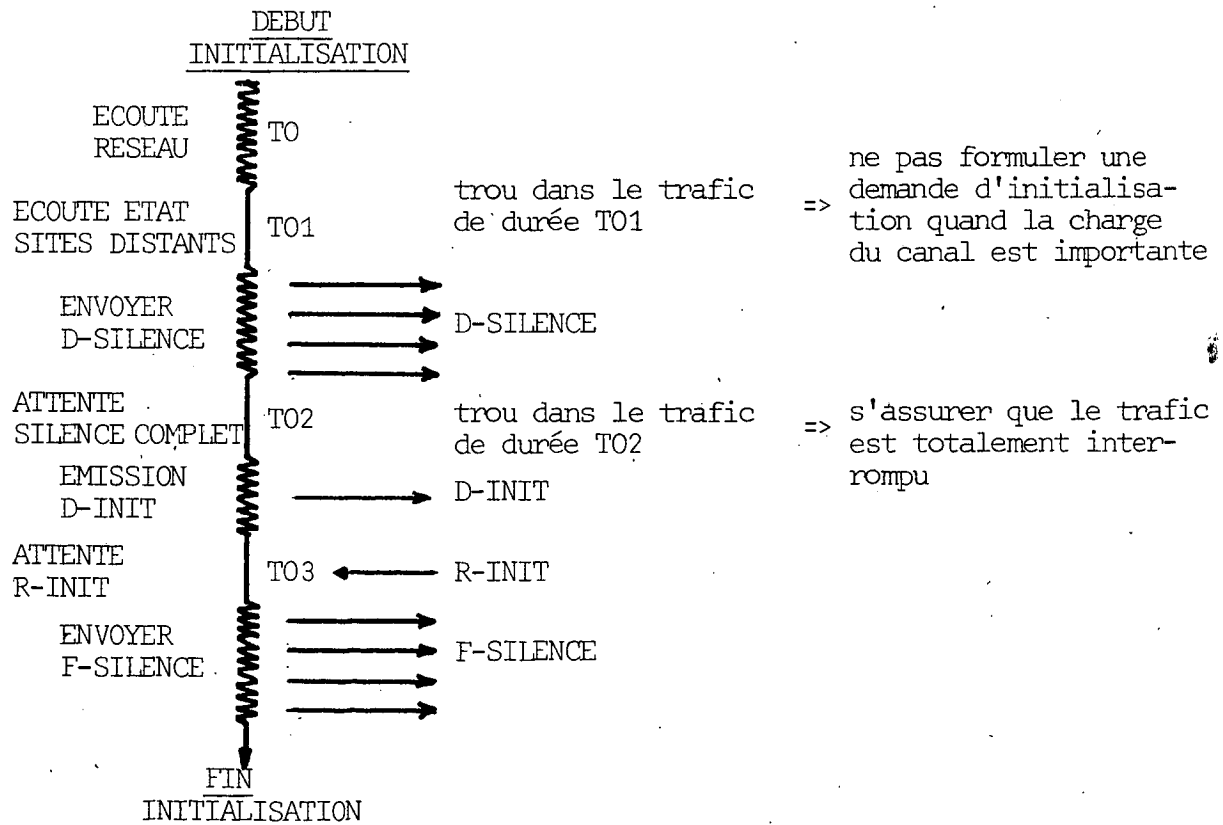
-Le time-out T02 permet de s'assurer que le trafic est totalement interrompu c'est-à-dire que tous les buffers émission sur le réseau pleins lors de la réception de la primitive D_SILENCE ont été vidés. Si le serveur se trouvant dans la phase "Attente Silence Complet" n'a reçu aucun paquet au terme d'un délai égal au temps maximum de transfert d'un paquet de buffer émission à buffer réception, il peut en déduire que tous les buffers émission sont vides. Il suffit donc d'affecter à T02 la valeur du délai maximum de transfert d'un paquet sur le support de transmission.

-Par ailleurs, le time-out T0, permettant de déterminer l'état du système, est très long et sa durée est largement supérieure à celle de tous les autres réveils utilisés.

Il est donc possible de poser la relation:

$$T0 > T03 > T02$$

Le [Schéma. 10] montre le déroulement temporel d'une tentative d'insertion sur le réseau.



SCHEMA 10 : Diagramme temporel d'une tentative d'initialisation.

L'idée consiste à choisir $T01$ tel que l'on ait la relation:
 $T0 > T01 > T03 > T02$

En effet, en procédant de cette manière, on est sûr que si un site commence une tentative d'insertion alors qu'un autre serveur a déjà soumis ses primitives `D_SILENCE`, il ne parviendra pas jusqu'à la phase "Envoyer `D_SILENCE`" car il ne pourra pas observer sur le canal le temps mort de durée $T01$ précédant l'émission des primitives `D_SILENCE`.

-Traitement T8

Lorsqu'un serveur initialisé reçoit la primitive `D_SILENCE`, il suspend ses émissions afin de contribuer à l'établissement du silence sur le canal, silence nécessaire à la poursuite de la tentative d'insertion en cours. On a vu que les émissions sont libérées sur réception de la primitive `F_SILENCE`.

Mais il se peut que le serveur ayant soumis les primitives `D_SILENCE` soit victime d'une défaillance avant d'avoir émis les paquets `F_SILENCE`: c'est un cas de blocage infini des émissions.

Pour éviter de tels problèmes, on a défini un time-out `TO-ID`, de longue durée, enclenché sur réception de `D_SILENCE`.

Le déclenchement de `TO-ID` force la libération des émissions.

* Quelques remarques *

-On peut vouloir contrôler la durée d'une tentative d'insertion sur le réseau.

Il suffit d'utiliser un time-out `TO-IL`, très long, qui sera armé au commencement de l'initialisation. Son déclenchement forcera l'arrêt immédiat de la tentative d'insertion en cours.

-Les paquets soumis par un serveur en cours d'initialisation peuvent être reçus erronés par certains sites. Leur retransmission sera demandée par la suite. Or ces paquets n'étant pas numérotés (puisque le but de la démarche d'initialisation est d'acquérir la valeur à jour de la variable `NUMERO`), le serveur qui les a émis ne sait pas que cette requête lui est adressée.

Face à une telle situation, il est possible de mettre en oeuvre un mécanisme de numérotation "a posteriori" des paquets émis pendant une tentative d'initialisation. Il suffit d'attribuer à chaque paquet émis un numéro relatif que l'on range dans une table de numérotation provisoire.

A la réception du R_INIT, il ne reste plus qu'à calculer les numéros réels des paquets émis en fonction du numéro mentionné dans le R_INIT.

Il est important de signaler que lorsque la retransmission de l'un de ces paquets est demandée, il est fondamental de renvoyer un paquet "bouche-trou" et non le paquet avec son contenu initial.

-De ce qui a été dit précédemment, on conclut qu'une tentative d'initialisation peut suivre deux directions:

.L'INITIALISATION LOCALE ACTIVE où le serveur concerné formule explicitement la demande d'acquisition de la valeur NUMERO, D_INIT.

.L'INITIALISATION LOCALE PASSIVE qui correspond à l'écoute d'une initialisation locale active.

Ce cas se produit quand le serveur détecte une tentative d'initialisation plus avancée ou prioritaire par rapport à la sienne pendant les phases précédant l'émission de la primitive D_INIT.

Description de l'initialisation locale passive

Quatre possibilités se présentent:

a- Pendant les phases d'écoute ("Ecoute Réseau" ou "Ecoute Etat Sites Distants"), réception d'une primitive D_SILENCE.

b- Pendant les phases d'écoute, réception d'une primitive D_INIT.

c- Entre le moment où le serveur est autorisé à soumettre ses primitives D_SILENCE et l'établissement du silence complet sur le canal, réception d'une primitive D_SILENCE "prioritaire" (ce cas correspond à des demandes d'initialisation simultanées).

d- Pendant les phases d'écoute, réception d'une primitive R_INIT.

*Les cas a, b et c sont traités de manière analogue. Le serveur passe dans la phase "Ecoute Initialisation Distante" et essaie d'intercepter le R_INIT afin de réaliser sa propre insertion. Au début de cette phase, il est nécessaire d'armer le time-out TO-ID (dont on a déjà parlé), ceci au cas où le serveur émetteur de la primitive qui a provoqué le changement de phase serait victime d'une défaillance.

Trois événements peuvent alors provoquer le changement d'état du serveur:

-Réception de la primitive R_INIT

La tentative d'insertion dynamique sur le réseau est réussie: le serveur initialisé sa copie de la variable NUMERO de la valeur mentionnée dans le paquet R_INIT.

Cependant, il ne doit pas encore soumettre de paquets de données car le signal annonçant la fin de la tentative d'initialisation du serveur ayant soumis la demande D_INIT n'a pas été reçu (primitive F_SILENCE). Aussi est-il nécessaire de se mettre en attente de ce signal (phase "Fin Ecoute Initialisation Distant").

-Réception de la primitive F_SILENCE

L'occurrence de cet événement implique que le serveur n'a pas vu passer le paquet R_INIT.

Deux explications sont envisageables:

- .aucune primitive R_INIT n'a été diffusée.
- .la primitive R_INIT a été reçue erronée.

C'est un cas d'échec de la tentative d'initialisation: le serveur désarme le réveil TO-ID et renouvelle une tentative d'insertion sur le réseau (phase "Ecoute Réseau").

-Déclenchement du réveil TO-ID

La durée de ce time-out étant suffisamment longue pour laisser se dérouler une tentative d'initialisation même dans les conditions les plus défavorables, le serveur peut en conclure que l'absence de réception des primitives F_SILENCE a pour cause la défaillance du site qui aurait dû les émettre.

La primitive R_INIT n'ayant par ailleurs pas été reçue, ce cas correspond aussi à un échec de la tentative d'initialisation locale. Le serveur doit donc tenter à nouveau une insertion sur le réseau (phase "Ecoute Réseau").

*Le cas d entraîne le succès immédiat de la tentative d'insertion sur le réseau.

Le serveur met à jour son instance de la variable NUMERO. Cependant, il est nécessaire de reporter l'instant d'autorisation des émissions car le signal indiquant la fin de la tentative d'initialisation du serveur ayant formulé la demande D_INIT (primitive F_SILENCE) n'a pas encore été reçu. Le serveur passe dans la phase "Fin Ecoute Initialisation Distant" et arme le réveil TO-ID.

Trois évènements peuvent provoquer la transition du serveur dans l'état de fonctionnement normal:

-Réception de la primitive F_SILENCE

La tentative d'insertion du serveur ayant soumis la demande D_INIT est achevée.

-Réception d'un paquet de données

A ce stade d'une tentative d'insertion, si l'on reçoit un paquet de données sans avoir reçu au préalable la primitive F_SILENCE, on peut en conclure à la défaillance du serveur qui aurait dû émettre cette primitive.

Le fait de recevoir un paquet de données implique que le réveil TO-ID s'est déclenché sur au moins un site qui avait suspendu ses émissions afin de permettre l'établissement du silence nécessaire au déroulement de la tentative d'insertion.

Certains serveurs ayant repris leurs émissions, il n'y a aucune raison de vouloir maintenir leur suspension plus longtemps (au niveau du serveur considéré).

-Déclenchement du time-out TO-ID

Ce time-out étant très long, on peut là-encore en déduire la défaillance du serveur qui aurait dû émettre les primitives F_SILENCE.

Le silence établi sur le canal n'ayant plus besoin d'être maintenu, le serveur peut passer dans l'état de fonctionnement normal et commencer ses émissions.

Le [Schéma 11] reprend en détail les traitements relatifs à l'initialisation locale en distinguant bien les deux aspects exposés: INITIALISATION LOCALE ACTIVE et INITIALISATION LOCALE PASSIVE.

```
*****
*                                     *
*   S C H E M A   11   *
*                                     *
*****
```

LEGENDE

=====

- Abréviations

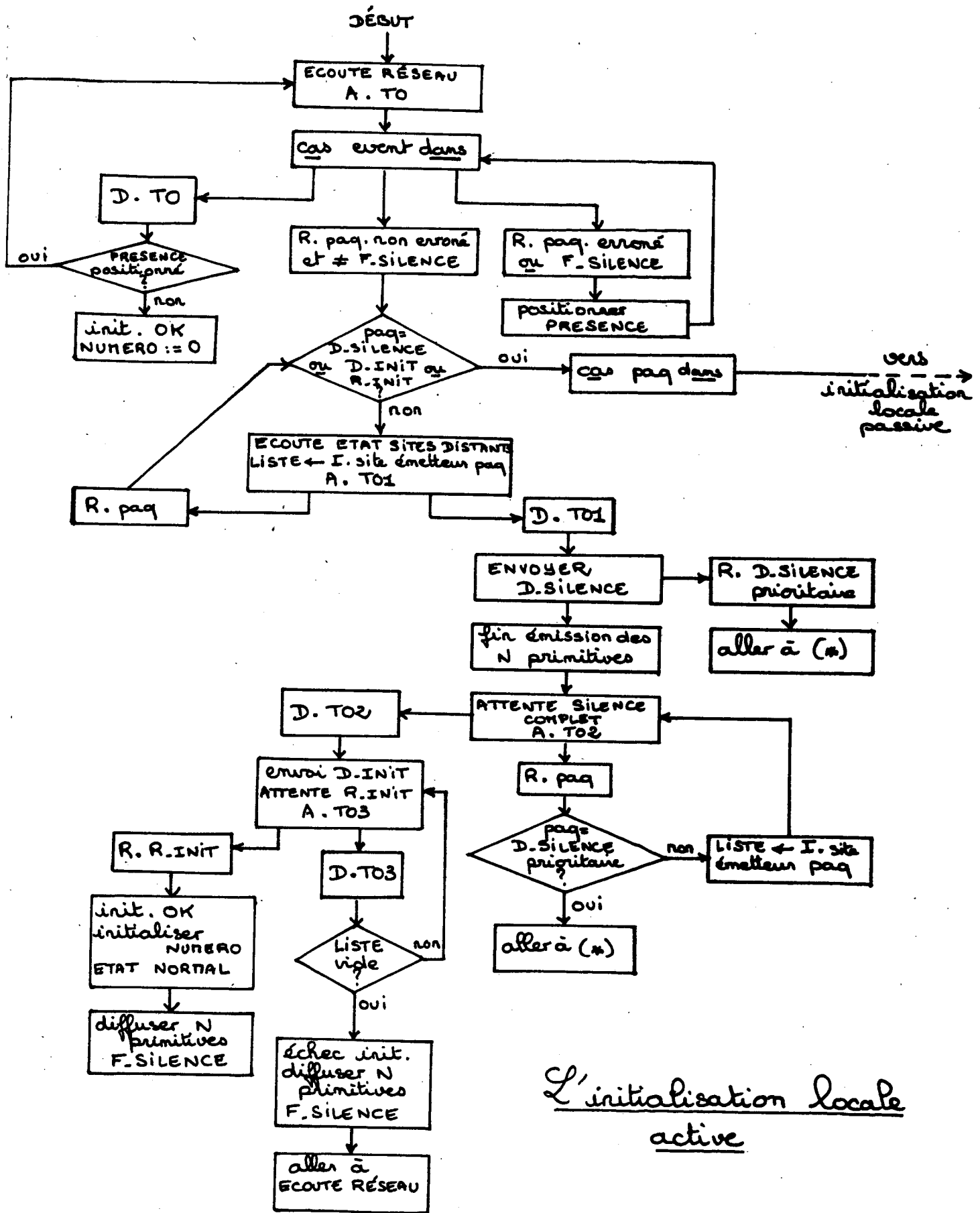
A.	armer le time-out
D.	déclenchement du time-out
I.	identification
R.	réception
<--	insérer dans
paq	paquet
init	initialisation
event	évènement

- Variables

.La variable PRESENCE représente le booléen témoin de l'activité d'au moins un site.

.L'identificateur LISTE désigne la liste des sites initialisés dont le serveur a détecté la présence.

L'ALGORITHME D'INSERTION DYNAMIQUE



L'initialisation locale active

3 - CONCLUSION

=====

L'algorithme décrit dans ce papier, bâti sur les hypothèses énoncées dans l'introduction, répond aux objectifs qui ont été fixés:

-L'insertion d'un site est une action peu prioritaire.

-La simultanéité de demandes d'initialisation est résolue de manière satisfaisante.

-L'algorithme tient compte de l'éventuelle occurrence d'erreurs de transmission ou de défaillances de processeurs.

Toutefois, cet algorithme ne garantit pas qu'une tentative d'insertion sur le réseau puisse être menée à terme au bout d'un temps fini.

Mais cette propriété ne doit pas être considérée comme une faille de l'algorithme étant donné qu'elle résulte du choix qui a été fait d'interdire l'insertion d'un nouveau serveur de communication sur le réseau lorsque le trafic est dense.

L'algorithme présente également l'inconvénient de solliciter un blocage des émissions pendant la phase d'initialisation d'un site.

Les traitements mis en oeuvre tendent à minimiser ce délai de suspension des échanges et de plus, en cas d'anomalie, des délais de garde ont été définis afin d'éviter le blocage infini des émissions.

Cet algorithme a été défini dans le cadre du développement d'une machine de transport adaptée à la mise en oeuvre d'applications bureautiques autour du réseau local DANUBE (cf [Danube 1] et [Danube 2]).

Pour de plus amples détails sur le logiciel de transport défini, se référer à [Gaucher-Cazalis] et à [Gaucher-Cazalis, Krier, Le Goff].

Dans une première phase, l'algorithme a été validé par simulation. De nombreuses expérimentations ont été pratiquées en faisant varier les différents paramètres du modèle de simulation et en les ajustant de manière à engendrer des

situations anormales.

Ces expérimentations ont été concluantes.

Ensuite, l'algorithme a été implémenté sur une carte micro-ordinateur à base de micro-processeur Z80. Les nombreux tests réalisés jusqu'à présent sont satisfaisants. De même que lors de la simulation, des anomalies de fonctionnement (défaillances de processeurs, erreurs de transmission) ont été provoquées au cours de ces tests.

Une description de l'algorithme d'insertion dynamique d'un site dans le cadre de cette étude est présentée dans [Le Coz].

REFERENCES

[Cornafion , 81]

Groupe Cornafion
"Systèmes Informatiques Répartis"
- Concepts et techniques -

Collection DUNOD
phase spécialité informatique, 1981

[Danube 1 , 80]

IRIA 1980
Projet Pilote KAYAK
REL 2.514.1 - Juillet 1980

"Description fonctionnelle du réseau
expérimental DANUBE"

[Danube 2 , 79]

LOGABAX 1979
J.P. SIBILLE - Mai 1979
Département Recherche et Développement

"Carte CTR80 / RL - Réseau DANUBE"

[Gaucher-Cazalis , 82]

S. Gaucher-Cazalis
Thèse de Docteur Ingénieur
Université de RENNES I
UER Mathématiques et Informatique
- Juillet 1982 -

"Conception d'une Machine de Transport
dans un environnement de réseau local"

[Gaucher-Cazalis, Krier , Le Goff , 80]

S. Gaucher-Cazalis ,
F. Krier ,
H. Le Goff
Publication Interne IRISA numéro 138
- Septembre 1980 -

"Définition d'un logiciel de transport
adapté à la mise en oeuvre d'applications
burotiques sur les réseaux locaux"

REFERENCES

[Le Coz , 82]

A. Le Coz
Rapport de stage D.E.A.
Université de RENNES I
UER Mathématiques et Informatique
- Juin 1982 -

"Insertion dynamique d'un processeur
dans un environnement de réseau local"

[Gaucher-Cazalis, Le Goff, Ollivier, 82]

S. Gaucher-Cazalis,
H. Le Goff ,
M. Ollivier
Congrès SEIR2
St Jacques de Compostelle, Espagne
Septembre 82.

"Architecture d'un serveur de communications multi-
points autour d'un réseau local"

